TechNote

QuickBuilder/QuickMotion Homing Example

34

Control Technology Corporation, Hopkinton, MA • 800.282.5008 • www.ctc-control.com

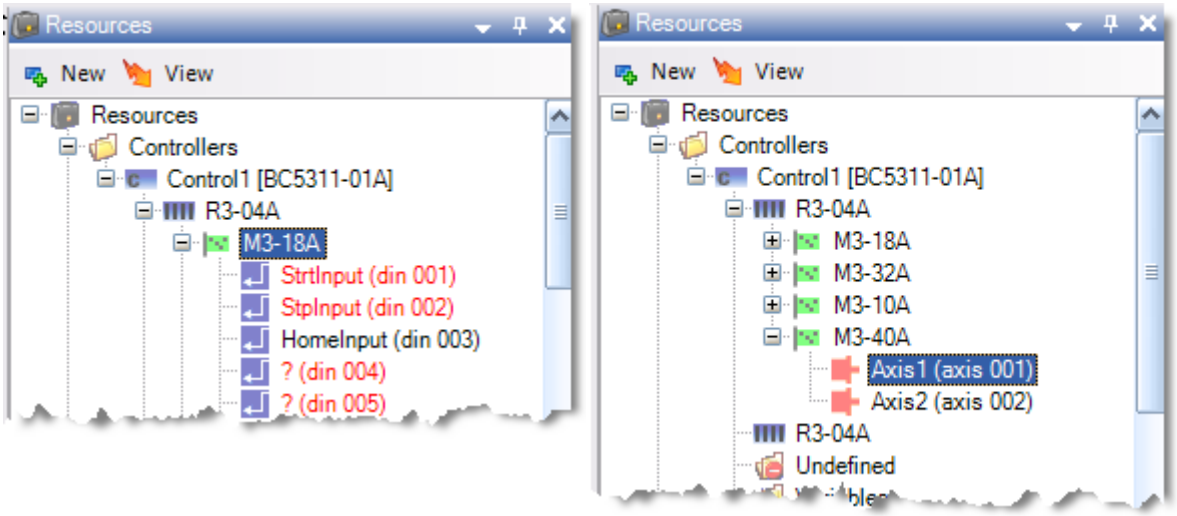
This document shows you a couple of the many ways you can set-up Homing using QuickMotion MSBs and QuickBuilder. This document assumes you already have a basic understanding of both QuickBuilder and QuickMotion. Refer to the QuickBuilder and QuickMotion Reference Guides if you are not already familiar with QuickBuilder and QuickMotion.

This program example uses two axes, each being homed using a slightly different method, which is often the case of what is required with a multi-axis system.

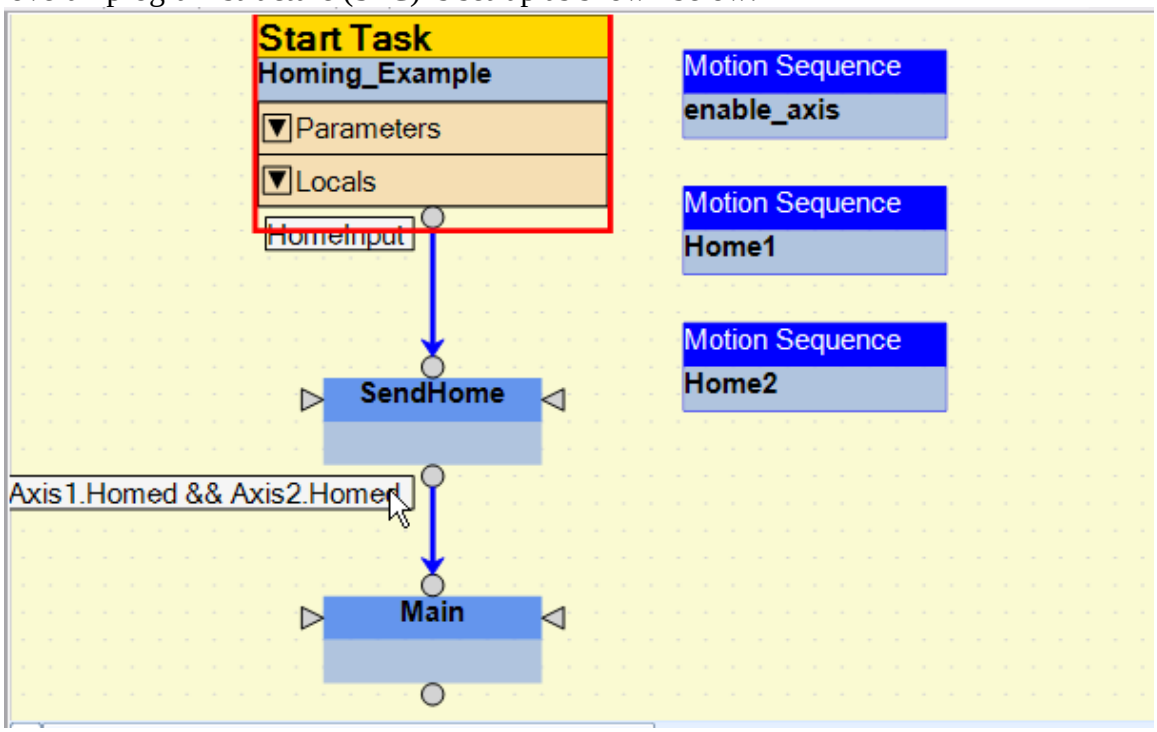
This program does the following:

1) Starts an enable_axis MSB for both Axis1 and Axis2 that enables both axes.
2) Waits for the HomeInput.
3) Starts the Homing MSBs for both Axis1 and Axis2.
4) Waits for feedback from the Axis variable (Homed) in the MSB to confirm both axes are homed.

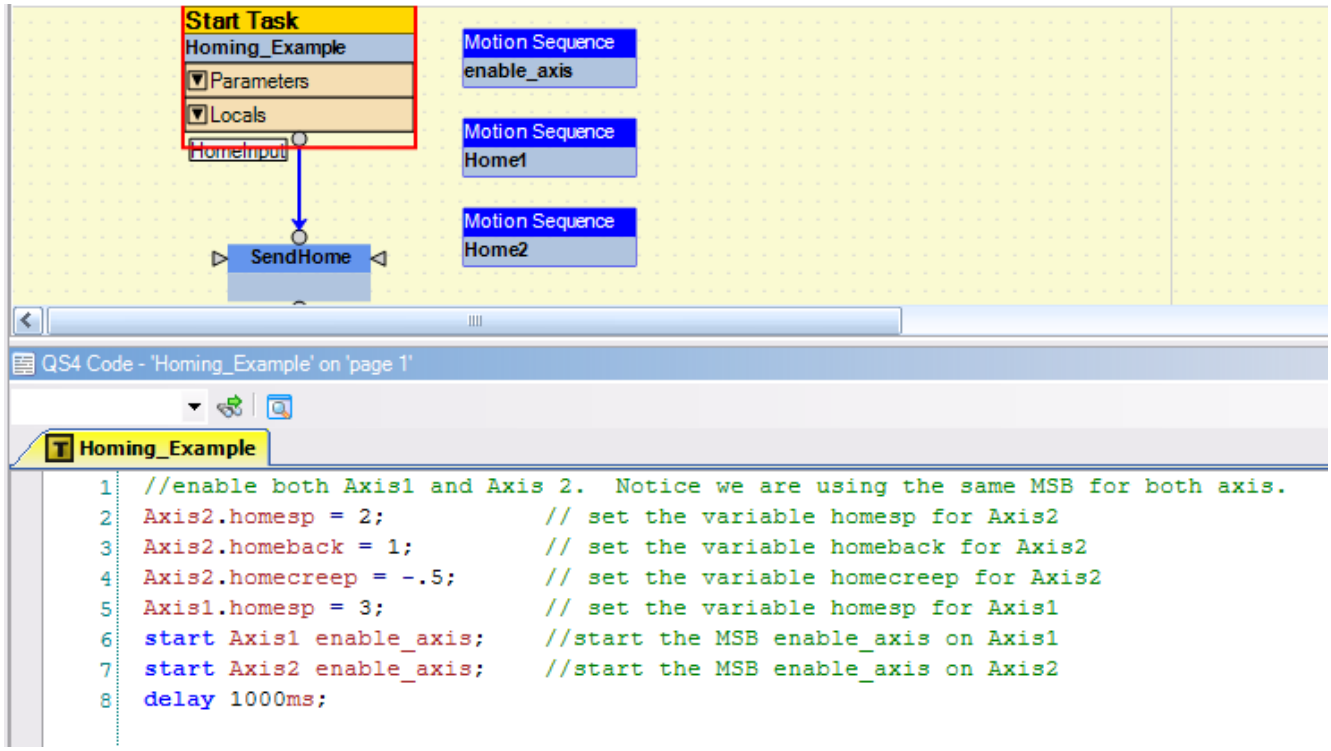
The system components (Resources) used in the program are shown below:



The overall program structure (SFC) is set-up as shown below:

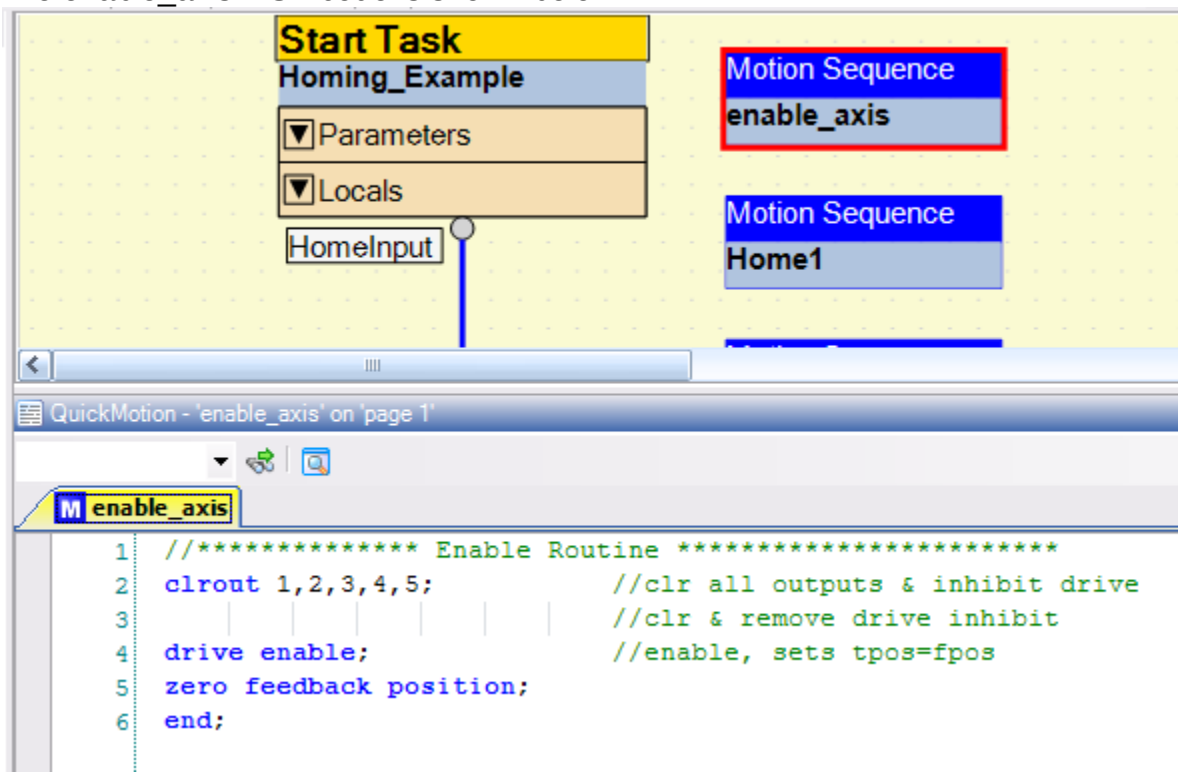


The initial Start Task enables the drives. You may or may not want to enable a drive right when the system is started, but we are doing so for simplicity in this program.

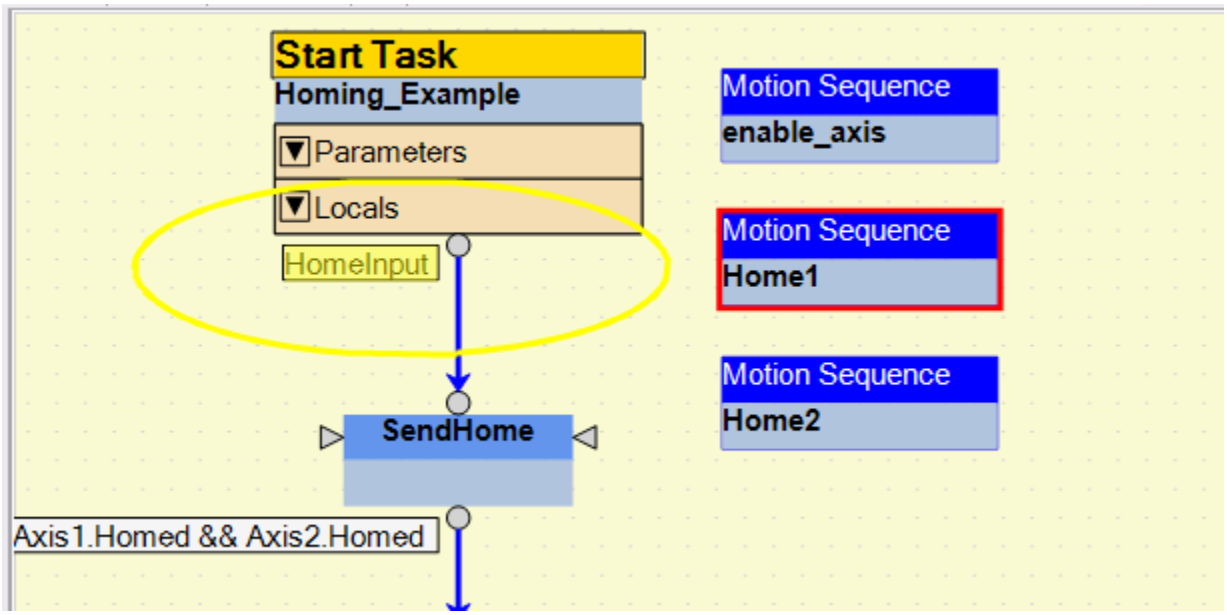


The code shown above will set the Axis2 and Axis 1 variables used in the Home1 and Home2 MSBs and start the enable_axis MSB for both Axis1 and Axis2.

The enable_axis MSB code is shown below.

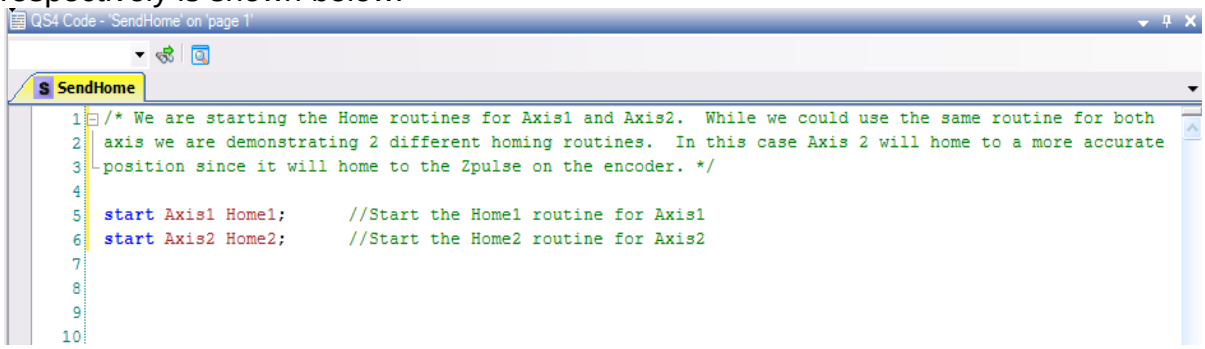


The drive enable command will both turn on the output associated with the driveenable property for the axis running that MSB and also begin to close the position loop for that axis.

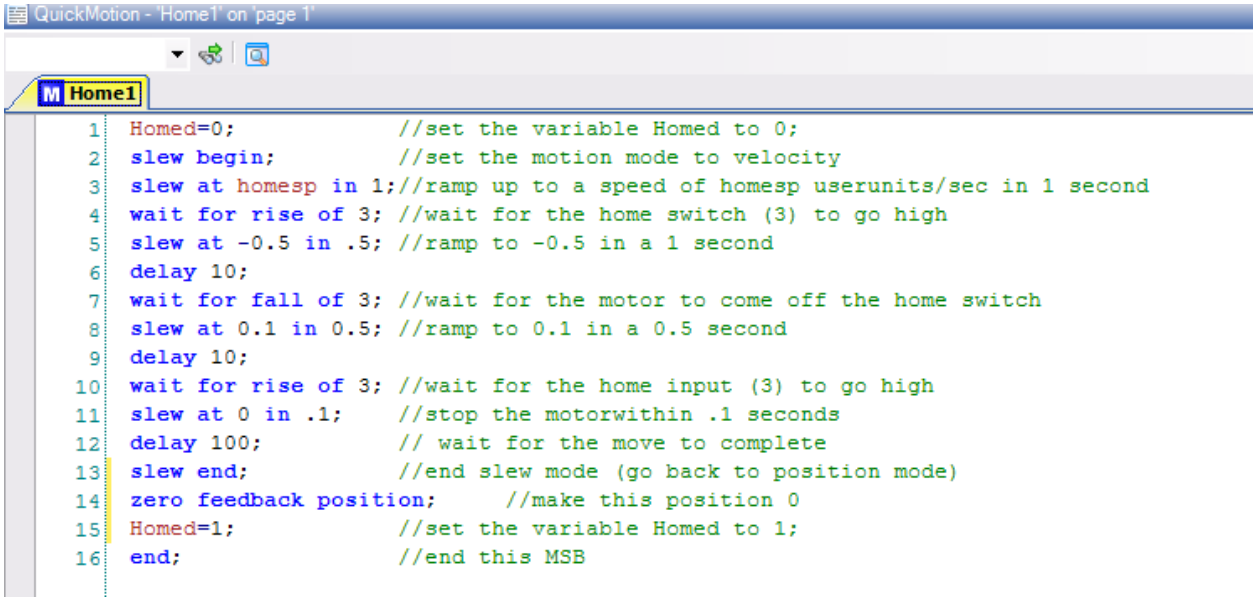


After the StartTask (Homing_Example) has started the enable_axis MSBs and after the delay, it will wait for the HomeInput to be true (active) as shown by the transition condition highlighted above.

Once the HomeInput is seen the SendHome Step will be executed. The code which simply starts the Home1 and Home2 MSBs for Axis1 and Axis2 respectively is shown below:



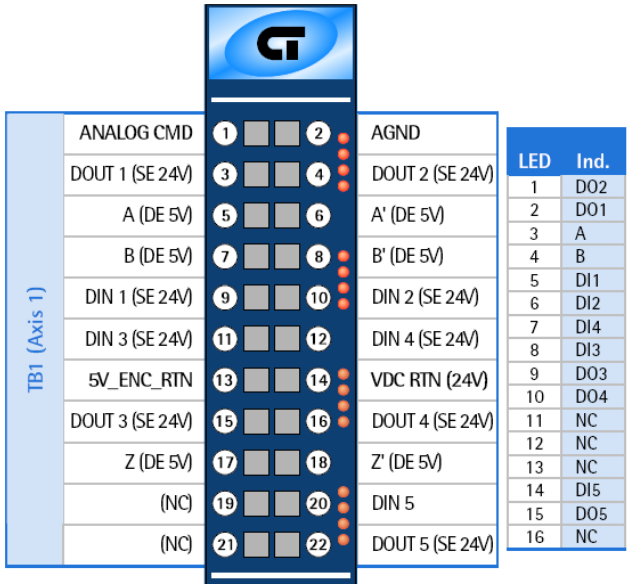
The code for the Home1 MSB is as follows:



Note: The initial speed used to approach the 3rd input on the M3-40A card is the value of the homesp variable This homesp variable is set in the QS4 Homing_Example StartTask code.

Note: The input 3 used for the homeswitch is not the HomeInput that is defined in resources to the M3-18A card, but the 3rd input (DI3) on the M3-40A card allocated to the Axis that is running the MSB (in this case Axis1).

See the M3-40A pin-out to the right.



LED	Ind.
1	DO2
2	DO1
3	A
4	B
5	DI1
6	DI2
7	DI4
8	DI3
9	DO3
10	DO4
11	NC
12	NC
13	NC
14	DI5
15	DO5
16	NC

The code for the Home2 MSB is as follows:

QuickMotion - 'Home2' on 'page 1'

Home2

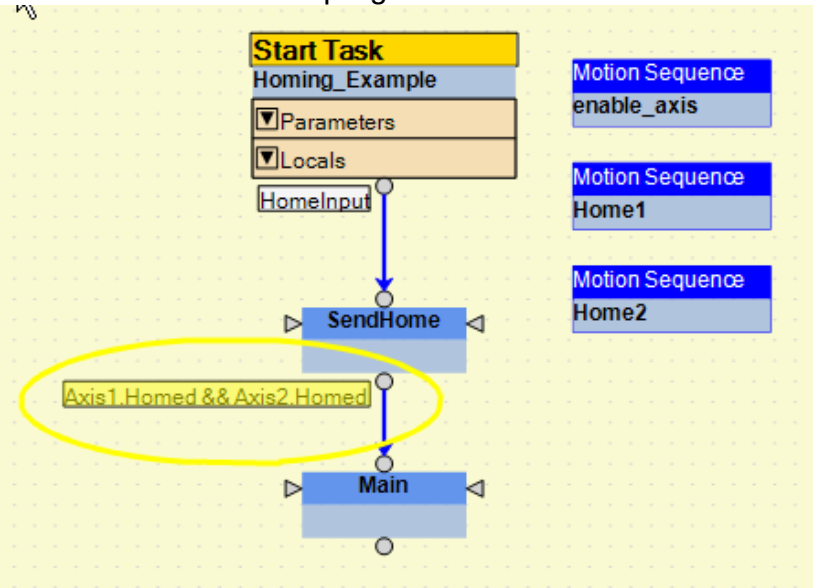
```
1  Homed=0;                 //set the variable Homed to 0;
2  slew begin;              //set the motion mode to velocity
3  slew at homesp in 1;     //ramp up to a speed of homesp userunits/sec in 1 second
4  wait for rise of 3;      //wait for the home switch (3) to go high
5  slew at -homebk in .5;   //ramp to -homebk userunits/sec in a 1 second
6  delay 10;
7  wait for fall of 3;      //wait for the motor to come off the home switch
8  slew at homecreep in 0.5; //ramp to homecreep userunits/sec in a 0.5 second
9  delay 10;
10 wait for rise of 3;      //wait for the home input (3) to go high
11 slew at 0 in .1;         //stop the motorwithin .1 seconds
12 delay 100;               // wait for the move to complete
13 slew end;                //end slew mode (go back to position mode)
14 move at indexrate to ZPULSE_POS;  //move to the zpulse in the positive direction
15                                   //at the rate of indexrate
16 wait for in position;    // wait for this move to complete
17 zero feedback position; //make this position 0
18 Homed=1;                //set the variable Homed to 1;
19 end;                    //end this MSB
```

This MSB is very close to what Home1 does, but it has an added move to the next Zpulse in the positive direction.

Note: Both the Home1 and Home2 MSB use a couple of the same variables. This is perfectly legal since each axis will have a unique set of variables even if they go by the same name (which is why we set both an Axis1.homesp and Axis2.homesp in the StartTask step). In fact it is necessary when referring to

an MSB variable within Quickstep code to designate the Axis for which you are setting the variable.

The last thing this example does is wait for the Axes' variables Homed to be true (=1) by the MSBs as shown below by the highlighted transition condition at which time the main program would start.



APPENDIX – MSB CODE

This appendix has the code shown in the steps and MSB's above but will allow you to cut and past into your documents for your convenience.

StartTask: Homing_Example Code:

```
//enable both Axis1 and Axis 2.  Notice we are using the same MSB for
both axis.
Axis2.homesp = 2;              // set the variable homesp for Axis2
Axis2.homeback = 1;                  // set the variable homeback for
Axis2
Axis2.homecreep = -.5;         // set the variable homecreep for Axis2
Axis1.homesp = 3;              // set the variable homesp for Axis1
start Axis1 enable_axis;       //start the MSB enable_axis on Axis1
start Axis2 enable_axis;       //start the MSB enable_axis on Axis2
delay 1000ms;
```

Step: SendHome Code:

```
/* We are starting the Home routines for Axis1 and Axis2.  While we
could use the same routine for both
```

```
axis we are demonstrating 2 different homing routines.  In this case
Axis 2 will home to a more accurate
position since it will home to the Zpulse on the encoder. */

start Axis1 Home1;            //Start the Home1 routine for Axis1
start Axis2 Home2;            //Start the Home2 routine for Axis2
```

MSB: enable_axis Code:

```
//************** Enable Routine ************************
clrout 1,2,3,4,5;             //clr all outputs & inhibit drive
                                        //clr & remove drive inhibit
drive enable;                       //enable, sets tpos=fpos
zero feedback position;
end;
```

MSB: Home1 Code:

```
Homed=0;                //set the variable Homed to 0;
slew begin;             //set the motion mode to velocity
slew at homesp in 1;//ramp up to a speed of homesp user units/sec in 1
second
wait for rise of 3;     //wait for the home switch (3) to go high
slew at -0.5 in .5; //ramp to -0.5 in a 1 second
delay 10;
wait for fall of 3; //wait for the motor to come off the home switch
slew at 0.1 in 0.5; //ramp to 0.1 in a 0.5 second
delay 10;
wait for rise of 3;     //wait for the home input (3) to go high
slew at 0 in .1;  //stop the motor within .1 seconds
delay 100;              // wait for the move to complete
slew end;               //end slew mode (go back to position mode)
zero feedback position;       //make this position 0
Homed=1;                //set the variable Homed to 1;
end;                    //end this MSB
```

MSB: Home2 Code:

```
Homed=0;                      //set the variable Homed to 0;
slew begin;                   //set the motion mode to velocity
slew at homesp in 1;    //ramp up to a speed of homesp userunits/sec in
1 second
wait for rise of 3;           //wait for the home switch (3) to go high
slew at -homebk in .5;  //ramp to -homebk userunits/sec in a 1 second
delay 10;
wait for fall of 3;     //wait for the motor to come off the home
switch
slew at homecreep in 0.5; //ramp to homecreep userunits/sec in a 0.5
second
delay 10;
wait for rise of 3;           //wait for the home input (3) to go high
slew at 0 in .1;        //stop the motor within .1 seconds
delay 100;                    // wait for the move to complete
slew end;                     //end slew mode (go back to position
mode)
```

```
move at indexrate to ZPULSE_POS;  //move to the zpulse in the positive
direction
                                                    //at the rate of
indexrate
wait for in position;   // wait for this move to complete
zero feedback position; //make this position 0
Homed=1;                      //set the variable Homed to 1;
end;                          //end this MSB
```